

PAT-NO: JP405204844A

DOCUMENT-IDENTIFIER: JP 05204844 A

TITLE: DIGITAL COMPUTER SYSTEM

PUBN-DATE: August 13, 1993

INVENTOR-INFORMATION:

NAME

PAN-RATZLAFF, RUBY Y

ASSIGNEE-INFORMATION:

NAME

DELL USA CORP

COUNTRY

N/A

APPL-NO: JP04210321

APPL-DATE: August 6, 1992

INT-CL (IPC): G06F013/38, G06F003/00

ABSTRACT:

PURPOSE: To automatically receive an arbitrary processor with one socket even if processors of plural types are different in correspondence relation between signals and pins by providing the processor with specific pins that engage the socket.

CONSTITUTION: The socket 10 is provided with a circuit needed to change the address of a specific signal to various pins according to the type of a processor in use. The output IGNNE*/MNI of PAL 18 is connected to a pin A15 of a processor 20. In this case, pins C14 and A13 supply a signal FERR or FERR1 to the PAL 18. The BIOS(basic input/output system) of the processor includes a program which detects the type of the processor when the computer system is powered on for the 1st time. An ID is supplied from the processor, and encoded by the BIOS 15 and stored in a register 16, and the BIOS 15 detects whether or not there is a coprocessor, and encodes discrimination information on the processor and stores it in the register 16.

COPYRIGHT: (C)1993,JPO

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-204844

(43)公開日 平成5年(1993)8月13日

(51)Int.Cl.⁵

G 0 6 F 13/38
3/00

識別記号

3 2 0 A 9072-5B
A 8323-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数8(全12頁)

(21)出願番号 特願平4-210321

(22)出願日 平成4年(1992)8月6日

(31)優先権主張番号 7 6 6 8 7 7

(32)優先日 1991年9月27日

(33)優先権主張国 米国(US)

(71)出願人 591134063

デル ユーエスエイ コーポレーション
アメリカ合衆国テキサス州、オースチン、
アーボアタム プールバード 9505

(72)発明者 ラビイ ワイ、パン - ラツラフ
追って補充致します。

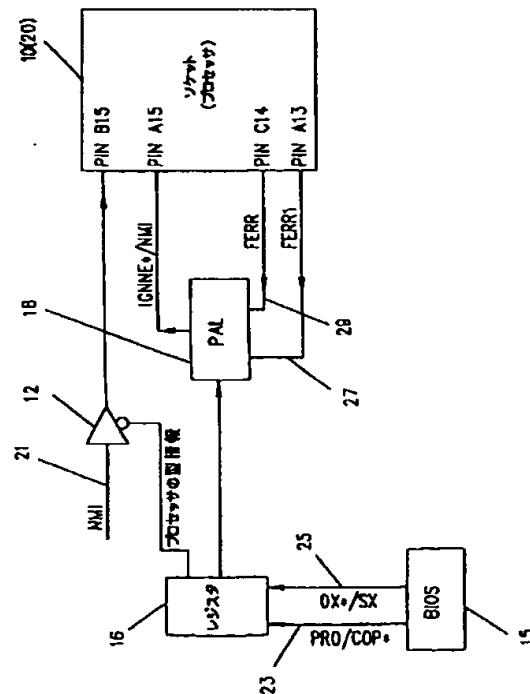
(74)代理人 弁理士 浅村 皓 (外3名)

(54)【発明の名称】 デジタルコンピュータシステム

(57)【要約】

【目的】 複数の型のプロセッサ間で信号とピンとの対応関係が異なっても、1個のソケットで任意のプロセッサを自動的に収容することができるようにする。

【構成】 本デジタルコンピュータシステムは、ファミリー中の任意のプロセッサを収容することのできるソケットを有する、ファミリーはそれぞれのプロセッサの基本入出力システムコードが共通であるとして定義する。各プロセッサでは特定の信号を通すためのピンが割り当てられており、ピンはソケットにはまるようになっている。少なくとも2種類のプロセッサ間では、少なくとも1個の特定の信号とピンとの対応関係が一致しない。このような不一致があると、プロセッサの型に応じて信号の宛先を適当なピンに自動的に変更する。



【特許請求の範囲】

【請求項1】 複数の型のプロセッサのうちの任意の1個を収容することができるプロセッサソケットを有するデジタルコンピュータシステムであって、該プロセッサは特定の信号を通し、ソケットに挿入されるためのピンを有し、該プロセッサ同士は共通の基本入出力システムコードを有すると共に、少なくとも1個の特定の信号を割り当てピンとが一致しないようになっている、デジタルコンピュータシステムにおいて、

(ア) プロセッサの型を検出して識別情報を供給するようにプログラムされているメモリと、

(イ) メモリとプロセッサとの間に接続されて、プロセッサが識別されたときに、少なくとも1個の特定の信号を別のピンに向けて供給するための論理回路と、を含むことを特徴とする、デジタルコンピュータシステム。

【請求項2】 請求項1記載の装置において、論理回路はプロセッサの識別情報を記憶するためのレジスタを含むことを特徴とする、デジタルコンピュータシステム。

【請求項3】 請求項2記載の装置において、論理回路は更に、レジスタの出力にตอบสนองするプログラマブルロジックアレイ (PAL) を含むことを特徴とする、デジタルコンピュータシステム。

【請求項4】 請求項3記載の装置において、論理回路は更に、レジスタの出力から供給されるゲート信号と、システムから供給される特定の信号とを受信するように接続されたゲートを含み、ゲートの出力は特定のピンに接続されていて、特定の信号が特定のピンに達するのを選択的に阻止するようになっていることを特徴とする、デジタルコンピュータシステム。

【請求項5】 請求項1記載の装置において、メモリは基本入出力システム (BIOS) を含むことを特徴とする、デジタルコンピュータシステム。

【請求項6】 請求項4記載の装置において、メモリは基本入出力システム (BIOS) を含むことを特徴とする、デジタルコンピュータシステム。

【請求項7】 請求項5記載の装置において、BIOSはプロセッサの型を読み取るためのプログラムと、コプロセッサの有無を検出するプログラムとを含むことを特徴とする、デジタルコンピュータシステム。

【請求項8】 請求項6記載の装置において、BIOSはプロセッサの型を読み取って、それを符号化し、そのコードをレジスタに記憶させるためのプログラムと、コプロセッサの有無を検出して、有無を符号化し、そのコードをレジスタに記憶させるためのプログラムとを含むことを特徴とする、デジタルコンピュータシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、プロセッサファミリー

に属する任意の1個のプロセッサを収容することのできるプロセッサソケットを有する、デジタルコンピュータシステムに関するものである。特に本発明はファミリー中の各種のプロセッサを自動的に収容することを可能にするものである。

【0002】

【従来の技術】 同じファミリー内のプロセッサでも、信号とピンの対応関係が異なる場合がしばしばある。すなわち、あるプロセッサのあるピンにある特定の信号が割り当てられているのに、別のプロセッサでは対応するピンに異なる信号が割り当てられている。

【0003】 これまでは、各プロセッサ毎にソケットを手動で調整するのが一般的であった。ジャンパ線をある位置から別の位置へと設けることによって、特定の信号を別のピンへ宛てるのである。この問題を解決する他の方法はスイッチを設けて、プロセッサの型に応じて特定の信号をあるピンから別のピンへ宛先を変えるのである。

【0004】 更に他の解決方法は、2個以上のソケットを使うことである。すなわち、あるソケットはファミリー中のあるプロセッサを受け入れるようにだけ結線しておき、第2、第3のソケットはファミリー中の他のプロセッサを受け入れるように結線するのである。このシステムでは、新しくプロセッサが実装されると、自動的に適当なプロセッサに切り替える回路が含まれることもある。

【0005】

【発明が解決しようとする課題】 本発明は、ジャンパ線を動かしたり、スイッチを投入したりせずに、使用者がファミリー内の所望のプロセッサを1個のソケットに挿入しさえすればよいシステムを提供する。動作は使用者に対して完全に明らかになっている。

【0006】

【課題を解決するための手段】 本デジタルコンピュータシステムは複数の型のプロセッサのうち任意の1個を収容することのできるソケットを有し、プロセッサは特定の信号を通し、ソケットとはまり合うためのピンを有している。各プロセッサは共通の基本入出力システムコードを有すると共に、少なくとも2種類のプロセッサ間では少なくとも1個の特定の信号とピンの割り当ての対応関係が異なっている。インテル社の80486DX、80486SXおよび80487SXから成るプロセッサファミリーは良い適用例である。もちろん他のファミリーのプロセッサにも適用することができる。

【0007】 基本入出力システムメモリ (BIOS) はプロセッサの型を読み取り、種類を符号化し、そのコードをレジスタに記憶させるためのプログラムを有する。BIOSはまた、コプロセッサの有無を検出し、有無を符号化し、コードをレジスタに記憶させるためのプログラムも有する。

【0008】プログラム可能なロジックのアレイ (PAL) をレジスタとソケットとの間に設ける。PALはあるピンに接続されていて、レジスタに記憶されている内容に依り、信号の宛先を別のピンに変える。ゲートを用いて、レジスタの内容に依り、プロセッサに信号が入るのを完全に阻止する。

【0009】本発明の目的は、複数の型のプロセッサ間で信号とピンの対応関係が異なっても、任意のプロセッサを自動的に受け入れるソケットをコンピュータシステムに提供することである。特定の信号が特定のピンに割り当てられており、共通の基本入出力システムコードを有してするプロセッサを対象としている。以下の詳細な説明により本目的および他の目的を明らかにする。

【0010】

【実施例】本発明によれば、単にプロセッサをソケットに挿入するだけで、使用者がプロセッサファミリー中の任意のプロセッサを交換することができるようになる。使用者はジャンパを挿入する必要もなければ、スイッチを入れる必要もない。指定された信号と割り当てられたピンとが合わないので、プロセッサのソケットを変更を加えることが必要であるが、それは使用者に対して完全に明らかである。以下この結果を得るために用いられるシステムを詳細に説明する。

【0011】図1～図5に本発明のソケットの略図を示す。ソケット10に付いている端子26は信号IGNN*

E を (図1)、端子27は信号FERR1* を (図1)、端子29は信号FERR* を (図1)、端子21は信号NMIを (図4) 運ぶ。信号PRO/COP* とDX* /SXはアンドゲート11で結合して、バッファ12を介して信号NMIの通過を制御する (図4)。「*」はロー信号レベルであることを示す。

【0012】図6はプロセッサとPALおよび関連部品を示すブロック図である。ソケット10には、使用されるプロセッサの型に依って、特定の信号をいろいろなピンに宛先を変えるのに必要な回路が付いている。PAL18の出力IGNNE* /NMIはプロセッサ20のピンA15に接続されている (プロセッサ20はソケット10に装着される)。ピンC14とA13はそれぞれ信号FERR* またはFERR1* をPAL18に供給する。本実施例ではPAL18の型は16R4-7である。PAL18は必要ときに、信号FERR* (80487SXと指定されたときにはFERR1* も) の宛先を変える。

【0013】本実施例では宛先を変更するのに必要な信号はNMIとIGNNE* のみである。プロセッサの型に依り、FERR* とFERR1* のどちらか一方が選ばれる。次表はこれらの信号とピンの対応を示す。

【0014】

【表1】ピンの相違の比較表ピン番号

信 号	80486DX	80486SX	80487SX
FERR*	C14	NC	A13
IGNNE*	A15	NC	A15
NMI	B15	A15	B15
NC: 接続なし			

【0015】プロセッサのBIOS15には、コンピュータシステムに最初に電源が入れたときに、プロセッサの型を検出するプログラムが含まれている。IDはプロセッサ自身により供給され、BIOS15により供給されるプログラムはそのIDを2ビットの2進コードに符号化して、それをレジスタ16に記憶させる。BIOS15はまた、コプロセッサがあるか否かを検出し、プロセッサの識別情報を加えるプログラムも有す ※

※。この情報は1ビットで符号化されてレジスタ16に記憶されるが、信号PRO/COP* として示してある。IDは信号DX* /SXとして示してあり、BIOS15から出力されてレジスタ16に記憶される。この識別化作業の全体のリストを以下に示す。

【0016】

【表2】

```

5
configure_processor_type:
    shld    eax,edx,24          ; プロセッサの改訂 ID は AE に入っている
; DX か SX かプロセッサの型を調べる -
;   もし改訂 ID が 486SX と適合すれば SX である。さもなければ DX とみなす
    mov     al,PORT_XXHIGH_INIT_VAL
    and     ah,11110000b        ; 下位ビットの改訂番号をマスクする
    cmp     ah,i486SX_REV_ID    ; もし CPU の改訂 ID が 486SX/487SX と
    je      its_an_sx           ; 同じであれば
    or      al,10000000b        ; DX プロセッサにセットする
    and     al,not 01000000b
    cmp     ah,i486_P23T_REV_ID
    je      its_P23T_P24        ; P23T and P24
    mov     ah,al
    jmp     finish_cpu_pin_setup
its_P23T_P24:
    out     PORT_XXHIGH,al      ; PORT_XXHIGH を記憶する
    mov     ah,al
    out     MATHBSY,al
    fninit
    fldcw   cs:unmasked_error_cw
    fstp    st(0)               ; FP をエラーにする
    fnstsw  word ptr cs:[0]
    WAFORIO
    WAFORIO
    mov     al,10b              ; IRR を流す
    out     0a0h,al
    in      al,0a0h
    out     MATHBSY,al          ; 懸案のエラーをクリアする
    fninit
    fldcw   cs:normal_cw
    test    al,20h              ; 未決の部分調べる
    jnz     finish_cpu_pin_setup ; もしエラーなら、P-24 は既に終了している
    or      ah,01000000b
    and     ah,not 10000000b
    jmp     finish_cpu_pin_setup
unmasked_error_cw    dw    0340h
normal_cw            dw    037fh
its_an_sx:
    and     al,00111111b        ; さもなくば SX である。ここで 486 か 487
                                ; かを調べる
; コプロセッサの存在の決定 -
;   1. メモリ位置を送込む
;   2. FPU をリセットする
;   3. FPU の制御語を同じメモリ位置に記憶させる
;   4. もし FPU が存在すれば、そのメモリ位置の内容が (FPU の制御語に)
;       変わっているだろう。
; 注意 - RAM がまだ確立されていないので、ROM の制御下でキャッシュメモリを
;        使う。

```

【0017】

* * 【表3】

```

7
mov     ebx, cr0
or      ebx, 60000000h      ; 486 のキャッシュメモリをオフにする
mov     cr0, ebx
invid

try_cpu_guess_again:      ; これ以外の方法を完全に理解しない限り、
                          ; このラベルの後、コードを変更しない。
                          ;
and     al, not PORT_XX_LOAD_RAM_M ; キャッシュプログラムモードをオンにする
or      al, 20h

lidt    cs:cpu_guess_tab
out     PORT_XXHIGH, al

; アドレスビット 3 - 1 は書き込み可能を意味する
; アドレスビット 2 - 1 キャッシュ不能を意味する

mov     bx, 0F000h
mov     ds, bx
xor     bx, bx
mov     cx, 1*4              ; F block, 16K or 32K units (32k
                          ; ユニットの場合 cx=2 で十分だが、cx=4 なら
                          ; 32k のみならず 16k でも有効であることに注意)

align   16
cache_486_init_loop_coproc:
mov     dx, ds:[bx]+0100b    ; 書き込み可能・キャッシュ可能
add     bx, 4000h
dec     cx
jnz     cache_486_init_loop_coproc

and     al, not 20h
out     PORT_XXHIGH, al
or      al, PORT_XX_LOAD_RAM_M ; キャッシュプログラムモードをオフにする
out     PORT_XXHIGH, al

mov     ebx, cr0
and     ebx, not 60000000h    ; 486 のキャッシュをオンにする
mov     cr0, ebx

mov     bx, ds:[0h]
mov     word ptr ds:[0], 0ffffh ; メモリ位置 0 の内容をキャッシュに読み込む
; -1 をキャッシュに書き込む
; FPU をリセットする
; FPU 制御語をローケーション 0 に記憶させる
fninit
fstcw   ds:[0h]
word ptr ds:[0], 0ffffh
cmp     word ptr ds:[0], 0ffffh
jbe     its_not_A_487         ; もしメモリが変わっていれば (FPU が発見)
; pro/cop ピンをコプロセッサにセットする

or      al, 01000000b
mov     ah, al
jmp     short finish_cpu_pin_setup

its_not_A_487:              ; ここでは pro/cop ピンをプロセッサにセットする
and     al, 00111111b
mov     ah, al

finish_cpu_pin_setup:      ; これ以外の方法を完全に理解しない限り、
                          ; このラベルの後にはコードを変えない
                          ;
lidt    cs:cpu_set_tab

mov     al, ah
out     PORT_XXHIGH, al

```

【0018】タイプ80486DXと80487SXの 40*NE* は80486DXと80487SXの場合にピン
 場合、NMI信号は図6に示すようにレジスタ16の出力
 力信号により制御されてバッファ12を通り、ピンB1
 5に供給される。しかしながら、もしプロセッサが80
 486SXならば、信号NMIは図6に示すようにPA
 L18を経由してピンA15に供給される。信号IGN*
 A15に加えられるが、80486SXの場合にはどこ
 にも接続されない。PAL18の動作のリストを以下に
 示す。
 【0019】
 【表4】

9

10

```

/** 入力 **/

Pin 1    = palclk2    /* プロセッサはクロックチップからクロックを得る */
Pin 2    = oscclk2    /* 上記クロックをコピーする */
Pin 3    = ferr       /* 486DX FERR* */
Pin 4    = ferrl_     /* 487SX FERR1* */
Pin 5    = pro_cop    /* 1 - 486SX を使用, 0 - 487SX を使用 */
Pin 6    = dx_sx      /* 1 - DX を使用, 0 - SX を使用 */
Pin 7    = resnpx     /* これ以上使用せず */
Pin 8    = bsy386_    /* coproc がビジーという信号をプロセッサに送る */
Pin 9    = !oe        /* 検査可能を示すピン. oe_ = 0 が有効 */
Pin 11   = !roe       /* レジスタの出力が検査可能であることを示すピン */
Pin 12   = nmi        /* システムからNMIを入力する */

/** 出力 **/

Pin 13   = errnpx_    /* コプロセッサエラーを 330 へ */
Pin 14   = !gnas_     /* */
Pin 15   = ferrd_     /* ferr* を1クロック遅らす */
Pin 16   = bsynpx     /* コプロセッサがビジー */
Pin 17   = clk2by2    /* プロセッサクロックを半分に */
Pin 18   = clk2syn     /* CLK2 に合わせる */
Pin 19   = !clksyn    /* CLK1 に合わせる */

/** 論理式 **/

clk2by2.d = !clk2by2 ;

clk2syn    = !oscclk2 & clk2by2 /* 両クロックが同じ遅延でオフになっている */
           # clksyn & clk2by2    /* ことを確認することにより、クロック2 */
           # clksyn & oscclk2 ;  /* とクロック1 からスキューを除く */

clk2syn    = !oscclk2 ;

errnpx_    = bsynpx_ ;

$define s0  'b'111
$define s1  'b'001
$define s2  'b'011
$define s3  'b'010
$define s4  'b'110
$define s5  'b'000
$define s6  'b'100
$define s7  'b'101

FIELD      npxerr = [ferrd_, bsynpx_, !gnas_];

SEQUENCEZ  npxerr {

present    s0
  if !nmi & pro_cop & !dx_sx next s4 ; /* 486SX の NMI が支配している */
  if !ferr_ & dx_sx next s1 ;
  if !ferrl_ & !dx_sx & !pro_cop next s1 ;
  default next s0 ;

```

【0020】

* * 【表5】

```

11
present s1
  if ferr_ & dx_sx next s0 ;          /* もしエラーがなくなっていれば戻る */
  if ferr1_ & !dx_sx & !pro_cop next s0 ;
  if pro_cop & !dx_sx next s4 ;        /* もし486SXならば、S4に行く */
  default next s2 ;

present s2
  if bsy386 next s3 ;                  /* ポート F0 に書きこまれてなければビジーにならない */
  if ferr_ & dx_sx next s0 ;
  if ferr1_ & !dx_sx & !pro_cop next s0 ;
  if pro_cop & !dx_sx next s4 ;
  default next s2 ;

present s3
  if ferr_ & dx_sx next s0 ;
  if ferr1_ & !dx_sx & !pro_cop next s0 ;
  if pro_cop & !dx_sx next s4 ;
  default next s3 ;

present s4
  if nmi & pro_cop & !dx_sx next s0 ;  /* 486SX の NMI を待つ */
  if dx_sx next s0 ;
  if !dx_sx & !pro_cop next s0 ;
  default next s4 ;

present s5
  next s0 ;

present s6
  next s0 ;

present s7

  next s0 ;

}

clksyn.oe = oe_ ;
clk2syn.oe = oe_ ;
arrnpx.oe = oe_ ;

```

【0021】80486SX型のプロセッサがソケット10に実装されているとしよう。この場合、信号NMIはゲート12によって阻止され、ソケット10のピンB15に達しない。信号NMIはPAL18を通して、表1に示した正しいピンであるA15に供給される。これがこの場合に必要となる唯一の宛先である。80487SX型のプロセッサが実装されているとき、NMI信号はバッファ12を通してソケット10のピンB15に達する。信号IGNNE*はPALからピンA15に供給される。ピンA13は走査されると、信号FERR1*を示す。

【0022】80486DX型のプロセッサが実装されると、信号NMIはバッファ12を通してピンB15に加えられる。信号IGNNE*はPAL18を介してピンA15に加えられる。ピンC14は走査されると信号*50

*FERR*を示す。このように、使用者は別の手続きや注意を要せずに、同じファミリーに属する少なくとも3種類のプロセッサを1個のソケットに実装することができる。

【0023】以上の説明は本発明の原理を説明するための例示にすぎない。当業者なら請求の範囲の欄に記載された本発明の範囲を逸脱することなく、多くの変形、修正例を容易に考えつくであろう。

【図面の簡単な説明】

【図1】本発明のソケットの略図。

【図2】本発明のソケットの略図。

【図3】本発明のソケットの略図。

【図4】本発明のソケットの略図。

【図5】本発明のソケットの略図。

【図6】プロセッサ、PAL、および関連部品のプロッ

ク図。

【符号の説明】

10 ソケット

12 バッファ

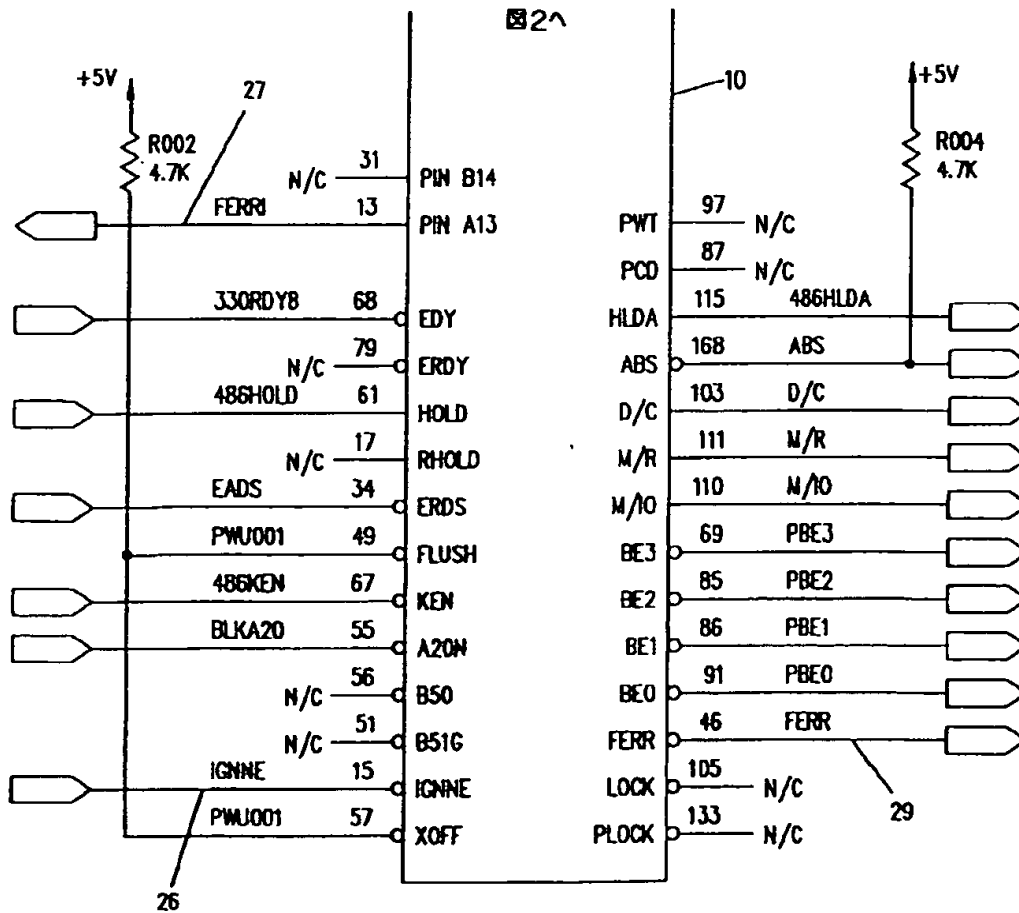
15 BIOS

16 レジスタ

18 PAL

20 プロセッサ

【図1】



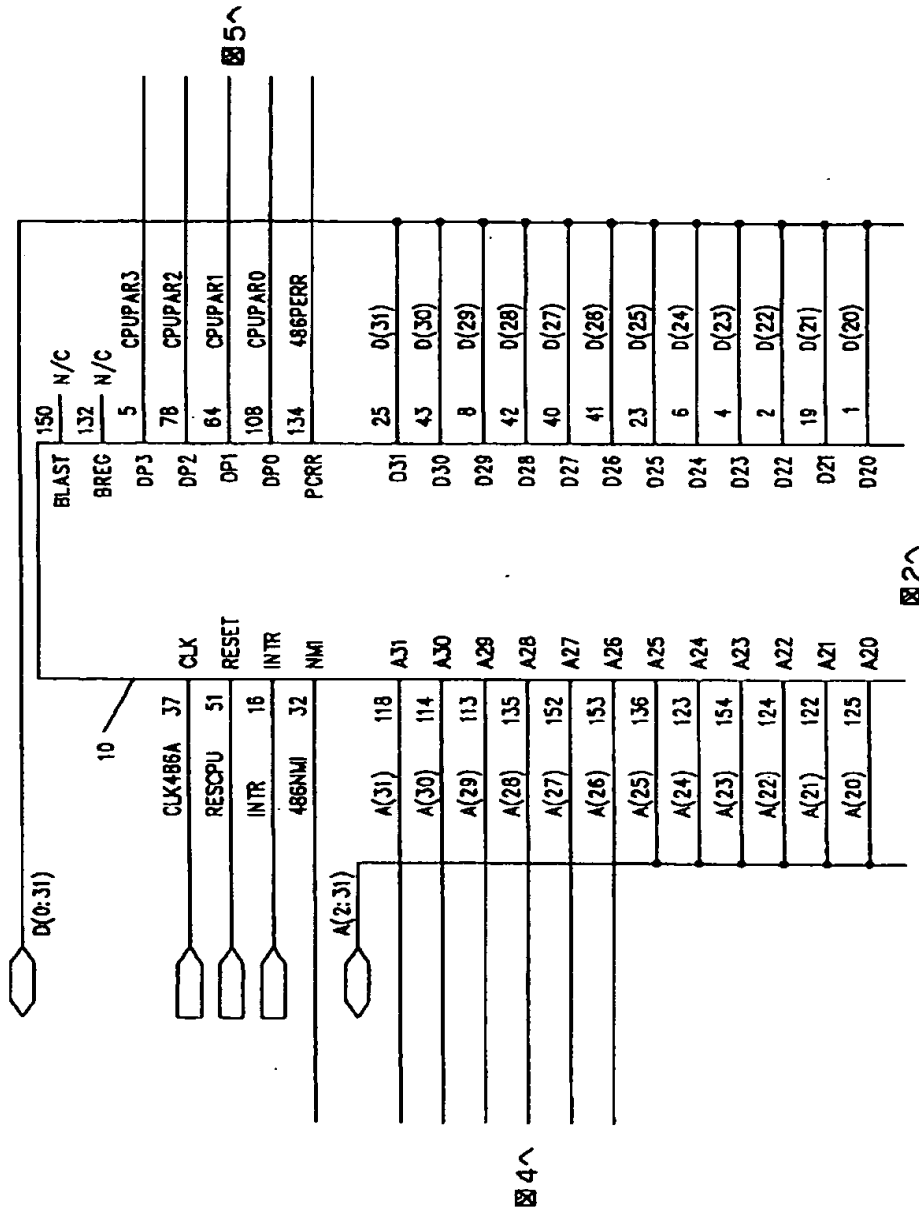
【図2】

図3へ

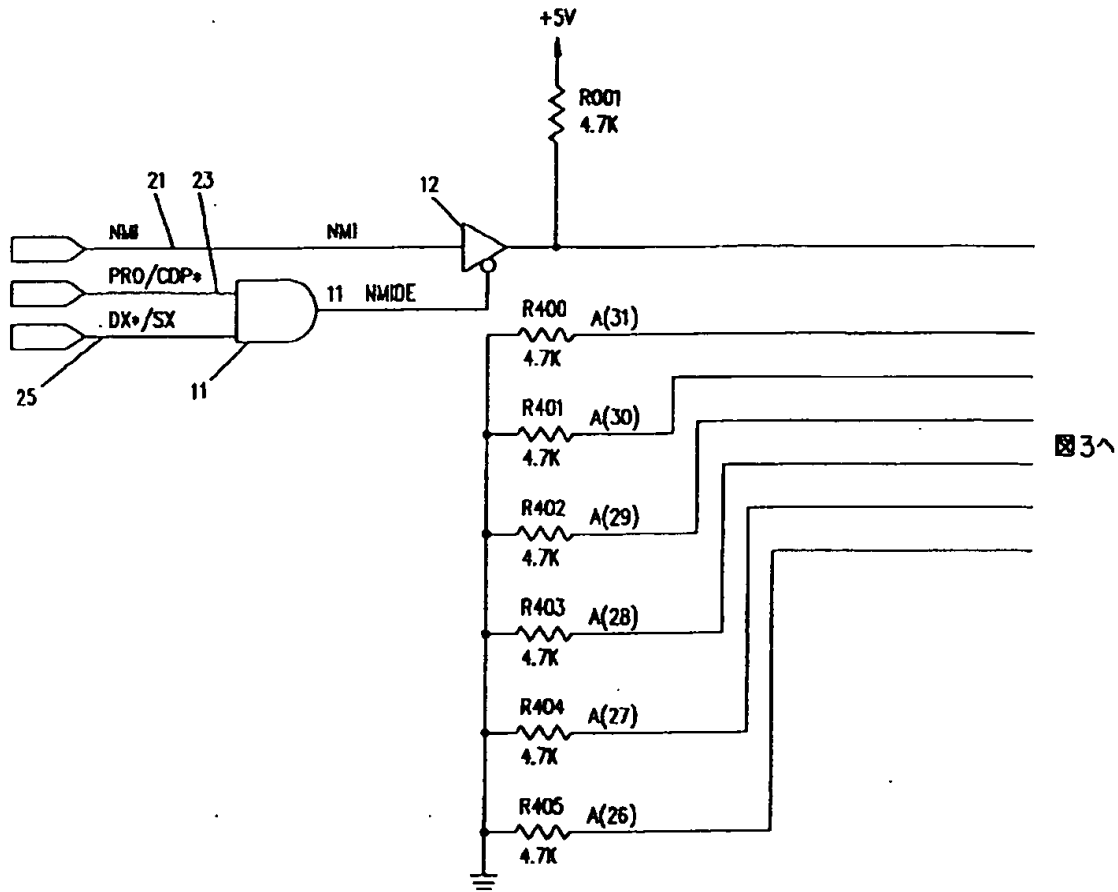
A(19)	121	A19	D19	18	D(19)
A(18)	139	A18	D18	36	D(18)
A(17)	120	A17	D17	54	D(17)
A(16)	126	A16	D16	84	D(16)
A(15)	141	A15	D15	66	D(15)
A(14)	156	A14	D14	90	D(14)
A(13)	127	A13	D13	53	D(13)
A(12)	158	A12	D12	72	D(12)
A(11)	146	A11	D11	35	D(11)
A(10)	164	A10	D10	60	D(10)
A(9)	128	A9	D9	52	D(9)
A(8)	147	A8	D8	65	D(8)
A(7)	130	A7	D7	96	D(7)
A(6)	166	A6	D6	95	D(6)
A(5)	129	A5	D5	83	D(5)
A(4)	167	A4	D4	102	D(4)
A(3)	149	A3	D3	77	D(3)
A(2)	131	A2	D2	106	D(2)
			D1	107	D(1)
			D0	112	D(0)

図1へ

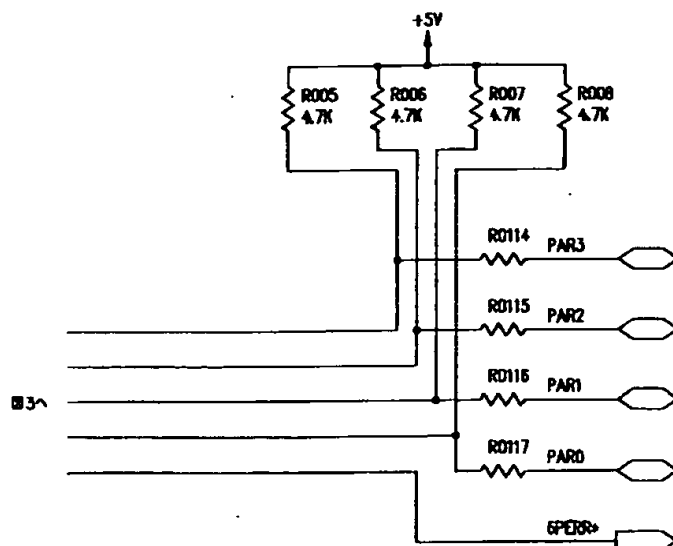
【図3】



【図4】



【図5】



【図6】

